Multi-robot Shaping of Deformable Objects based on Mechanical Simulation

GT SYSME day - Tarbes, September 30th, 2022

Adrien Koessler, associate professor

Université de Lorraine, Arts et Métiers Institute of Technology, LCFC,
HESAM Université, F-57070 Metz, France

People and Projects

- **Nicolas Roca Filella**, Institut Pascal, Université Clermont Auvergne, Michelin
- Prof. **Chedli Bouzgarrou**, Institut Pascal, Clermont Auvergne INP
- Dr. **Juan Antonio Corrales Ramón**, CiTIUS Lab, Uni. Santiago de Compostela
- **Laurent Lequièvre**, Institut Pascal, CNRS
- Dr. **Paul Baksic**, iCube, Université de Strasbourg
- Dr. **Hadrien Courtecuisse**, iCube, Inria Grand Est
- Prof. **Thibaut Raharijaona**, LCFC, Université de Lorraine

Interreg Sudoe COMMANDIA
European Regional Development Fund

UNIVERSITÉ Clermont Auvergne

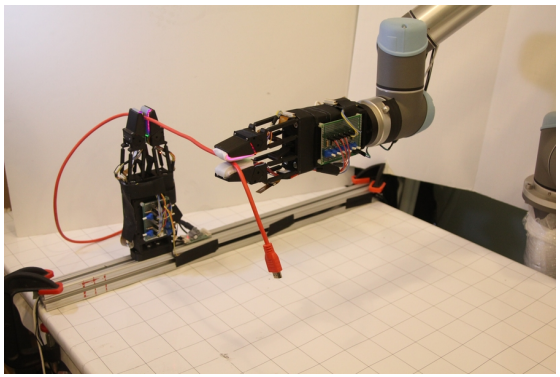
Interreg Grande Région | Großregion
Robotix-Academy
Fonds européen de développement régional | Europäischer Fonds für regionale Entwicklung

Contents

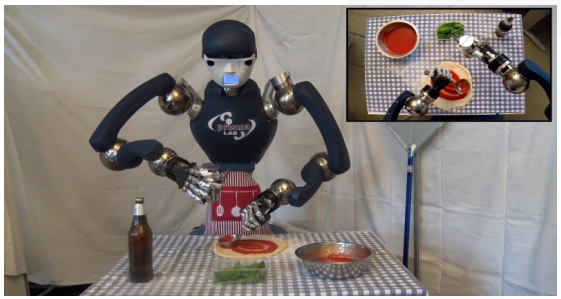
1. Introduction
2. Contribution outline
3. Methods
4. Experiments
5. Conclusion

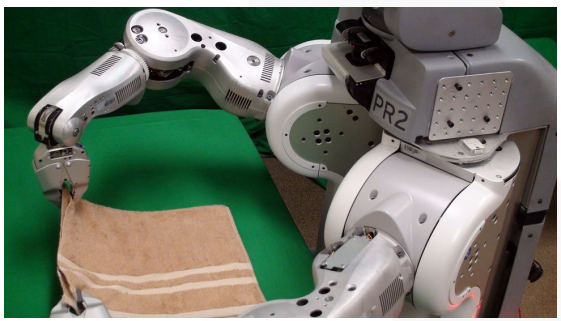
Context

Industry

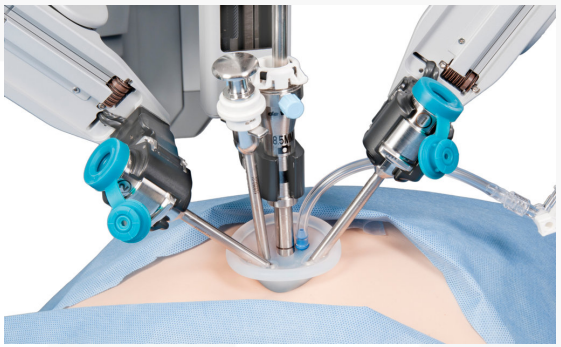


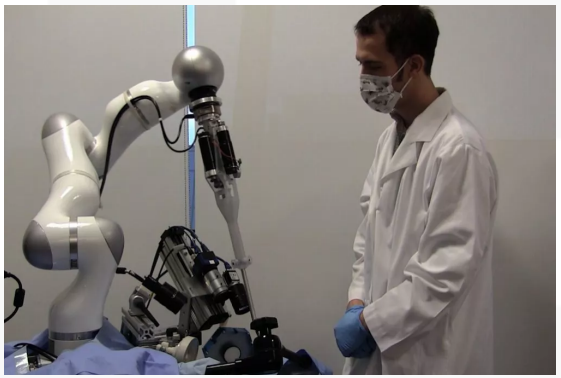
Household





Medical





Shape control

- Object deformation is an inconvenience treated as a perturbation

OR

- Object deformation is purposeful in system assembly and operation

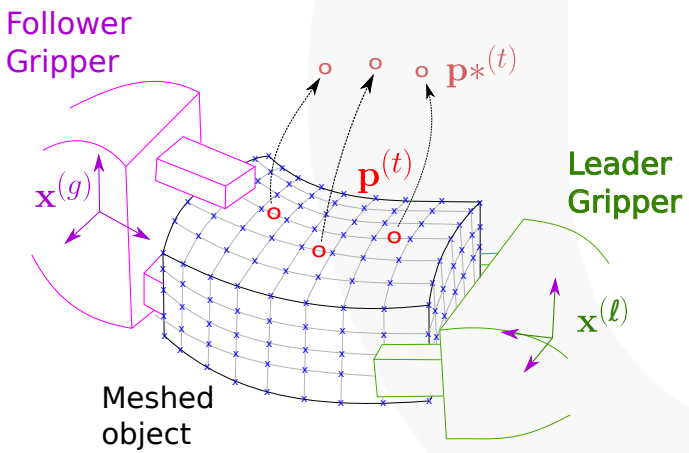
Shape control

- Object deformation is an inconvenience treated as a perturbation

OR

- Object deformation is purposeful in system assembly and operation

→ notion of **shape control**.



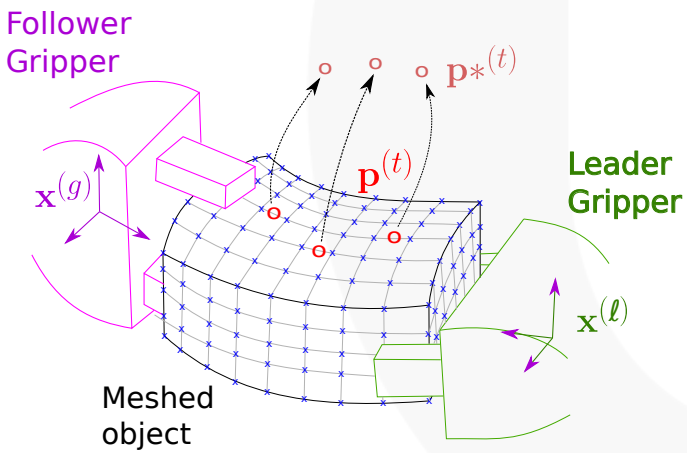
Shape control

- Object deformation is an inconvenience treated as a perturbation

OR

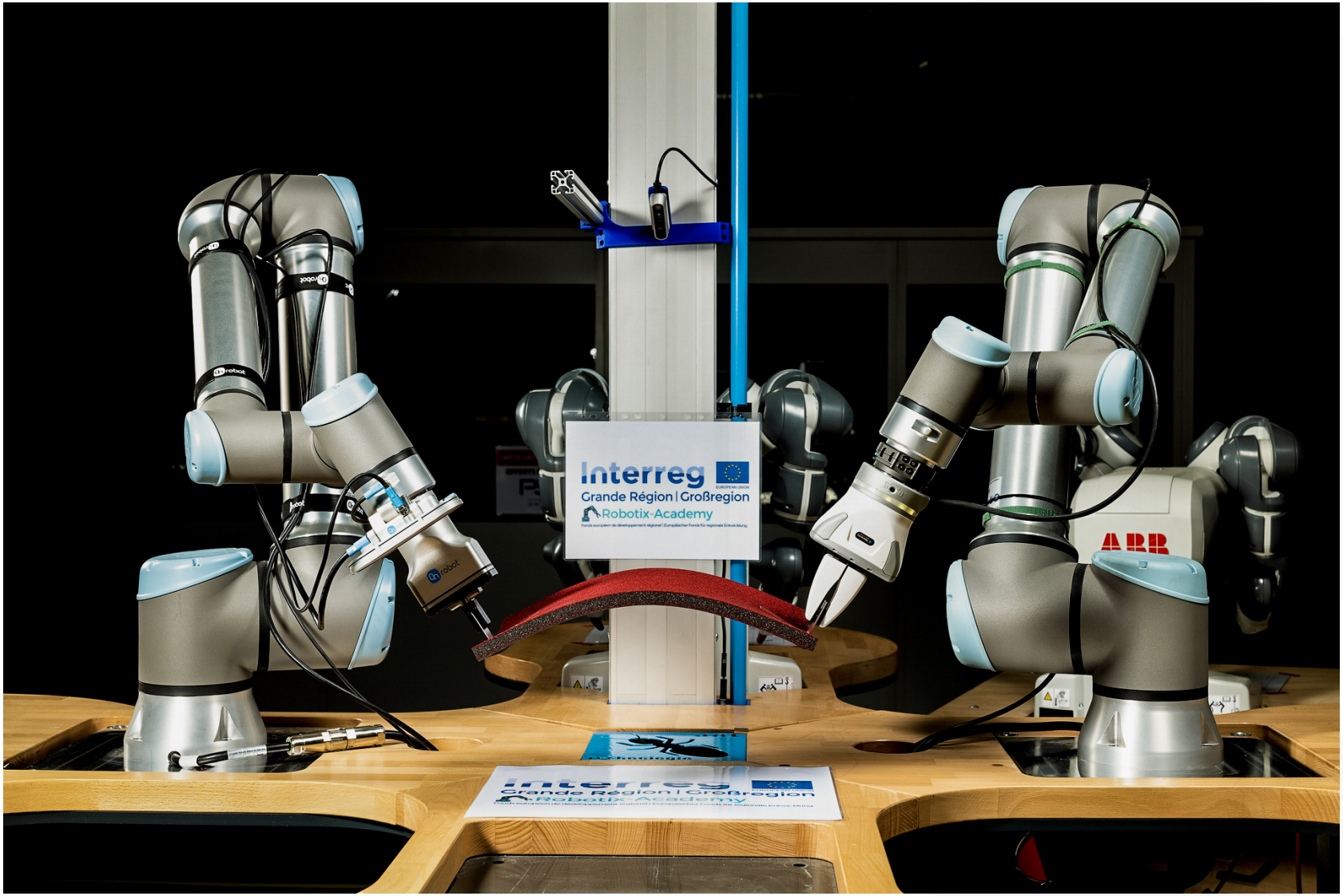
- Object deformation is purposeful in system assembly and operation

→ notion of **shape control**.



How to link $\mathbf{x}^{(g\ell)}$ and $\mathbf{p}^{(t)}$?

Manipulation example



Interaction Jacobian Matrix

The relation between $\mathbf{x}^{(g\ell)}$ and $\mathbf{p}^{(t)}$ is often linearized over time steps as:

$$\mathbf{p}_{k+1}^{(t)} = \mathbf{p}_{k}^{(t)} + \mathbf{J}(\mathbf{x}_{k+1}^{(g\ell)} - \mathbf{x}_{k}^{(g\ell)})$$

Interaction Jacobian Matrix

The relation between $\mathbf{x}^{(g\ell)}$ and $\mathbf{p}^{(t)}$ is often linearized over time steps as:

$$\mathbf{p}_{k+1}^{(t)} = \mathbf{p}_{k}^{(t)} + \mathbf{J}(\mathbf{x}_{k+1}^{(g\ell)} - \mathbf{x}_{k}^{(g\ell)})$$

$\mathbf{J}$ is the *Interaction Jacobian Matrix* of the system.

Interaction Jacobian Matrix

The relation between $\mathbf{x}^{(g\ell)}$ and $\mathbf{p}^{(t)}$ is often linearized over time steps as:

$$\mathbf{p}_{k+1}^{(t)} = \mathbf{p}_{k}^{(t)} + \mathbf{J}(\mathbf{x}_{k+1}^{(g\ell)} - \mathbf{x}_{k}^{(g\ell)})$$

$\mathbf{J}$ is the *Interaction Jacobian Matrix* of the system.

Estimating $\mathbf{J}_{inv}$ is of **crucial interest for shape control**
$\rightarrow$ Compute gripper motion to reach a given shape.

Literature overview

Desired shape formulation:

Jacobian $\mathbf{J}$ estimation:

[1][2][3] Zhu et al., RAS 142, 2021. Lagneau et al., RA-L 5(4), 2020. Laezza et al., ICRA 2021
[4][5][6] Berenson, IROS 2013. Aranda et al., IROS 2020. Shetab et al., RA-L 7(2), 2022
[7][8][9] Roca Filella et al., IROS 2022. Das et al., J. Int. Rob. Syst. 62, 2011. Duenser et al., IROS 2018

Literature overview

Desired shape formulation:

- Vision-based - $\mathbf{p}*^{(t)}$ are visual features, $p*_i^{(t)} = (u_i, v_i)$
- Model-based - $\mathbf{p}*^{(t)}$ are material points, $p*_i^{(t)} = (x_i, y_i, z_i)$
  Easily linked with camera calibration

Jacobian $\mathbf{J}$ estimation:

[1][2][3]Zhu et al., RAS 142, 2021. Lagneau et al., RA-L 5(4), 2020. Laezza et al., ICRA 2021
[4][5][6]Berenson, IROS 2013. Aranda et al., IROS 2020. Shetab et al., RA-L 7(2), 2022
[7][8][9]Roca Filella et al., IROS 2022. Das et al., J. Int. Rob. Syst. 62, 2011. Duenser et al., IROS 2018

Literature overview

Desired shape formulation:

- Vision-based - $\mathbf{p}*^{(t)}$ are visual features, $p*_i^{(t)} = (u_i, v_i)$
- Model-based - $\mathbf{p}*^{(t)}$ are material points, $p*_i^{(t)} = (x_i, y_i, z_i)$
  Easily linked with camera calibration

Jacobian $\mathbf{J}$ estimation:

- Online methods: Broyden update[1], adaptive control[2]

[123] Zhu et al., RAS 142, 2021. Lagneau et al., RA-L 5(4), 2020. Laezza et al., ICRA 2021
[456] Berenson, IROS 2013. Aranda et al., IROS 2020. Shetab et al., RA-L 7(2), 2022
[789] Roca Filella et al., IROS 2022. Das et al., J. Int. Rob. Syst. 62, 2011. Duenser et al., IROS 2018

Literature overview

Desired shape formulation:

- Vision-based - $\mathbf{p}*^{(t)}$ are visual features, $p*_i^{(t)} = (u_i, v_i)$
- Model-based - $\mathbf{p}*^{(t)}$ are material points, $p*_i^{(t)} = (x_i, y_i, z_i)$
  Easily linked with camera calibration

Jacobian $\mathbf{J}$ estimation:

- Online methods: Broyden update[1], adaptive control[2]
- Learned models: supervised learning, reinforcement learning[3]

[123] Zhu et al., RAS 142, 2021. Lagneau et al., RA-L 5(4), 2020. Laezza et al., ICRA 2021
[456] Berenson, IROS 2013. Aranda et al., IROS 2020. Shetab et al., RA-L 7(2), 2022
[789] Roca Filella et al., IROS 2022. Das et al., J. Int. Rob. Syst. 62, 2011. Duenser et al., IROS 2018

Literature overview

Desired shape formulation:

- Vision-based - $\mathbf{p}*^{(t)}$ are visual features, $p*_i^{(t)} = (u_i, v_i)$
- Model-based - $\mathbf{p}*^{(t)}$ are material points, $p*_i^{(t)} = (x_i, y_i, z_i)$
  Easily linked with camera calibration

Jacobian $\mathbf{J}$ estimation:

- Online methods: Broyden update[1], adaptive control[2]
- Learned models: supervised learning, reinforcement learning[3]
- Mechanical models:

Pseudo-physical models

- Diminishing rigidity[4]
- Position-based dynamics[5]
- As-rigid-as-possible[6]

Mechanical models

- Exact models (catenary)[7]
- Mass-spring-damper models[8]
- Finite element analysis[9]

[123] Zhu et al., RAS 142, 2021. Lagneau et al., RA-L 5(4), 2020. Laezza et al., ICRA 2021
[456] Berenson, IROS 2013. Aranda et al., IROS 2020. Shetab et al., RA-L 7(2), 2022
[789] Roca Filella et al., IROS 2022. Das et al., J. Int. Rob. Syst. 62, 2011. Duenser et al., IROS 2018

Contents

1. Introduction
2. Contribution outline
3. Methods
4. Experiments
5. Conclusion

Criteria for method choice

Advantages of vision:

- Readily available shape error signal $\rightarrow$ **feedback control**
- Robust to transformation errors (eg. camera calibration)

Criteria for method choice

Advantages of vision:

- Readily available shape error signal $\rightarrow$ **feedback control**
- Robust to transformation errors (eg. camera calibration)

Advantages of mechanical models:

- **General-purpose** in terms of geometries and materials
- Models may include external interactions

Criteria for method choice

Advantages of vision:

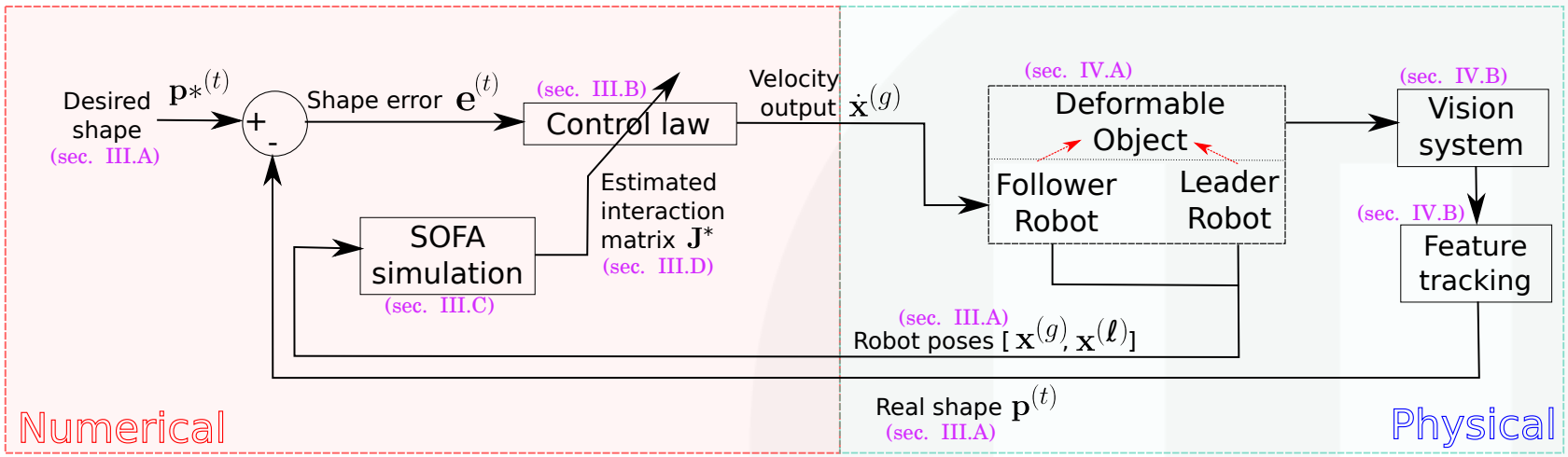
- Readily available shape error signal → **feedback control**
- Robust to transformation errors (eg. camera calibration)

Advantages of mechanical models:

- **General-purpose** in terms of geometries and materials
- Models may include external interactions

→ Merged in the proposed contribution.

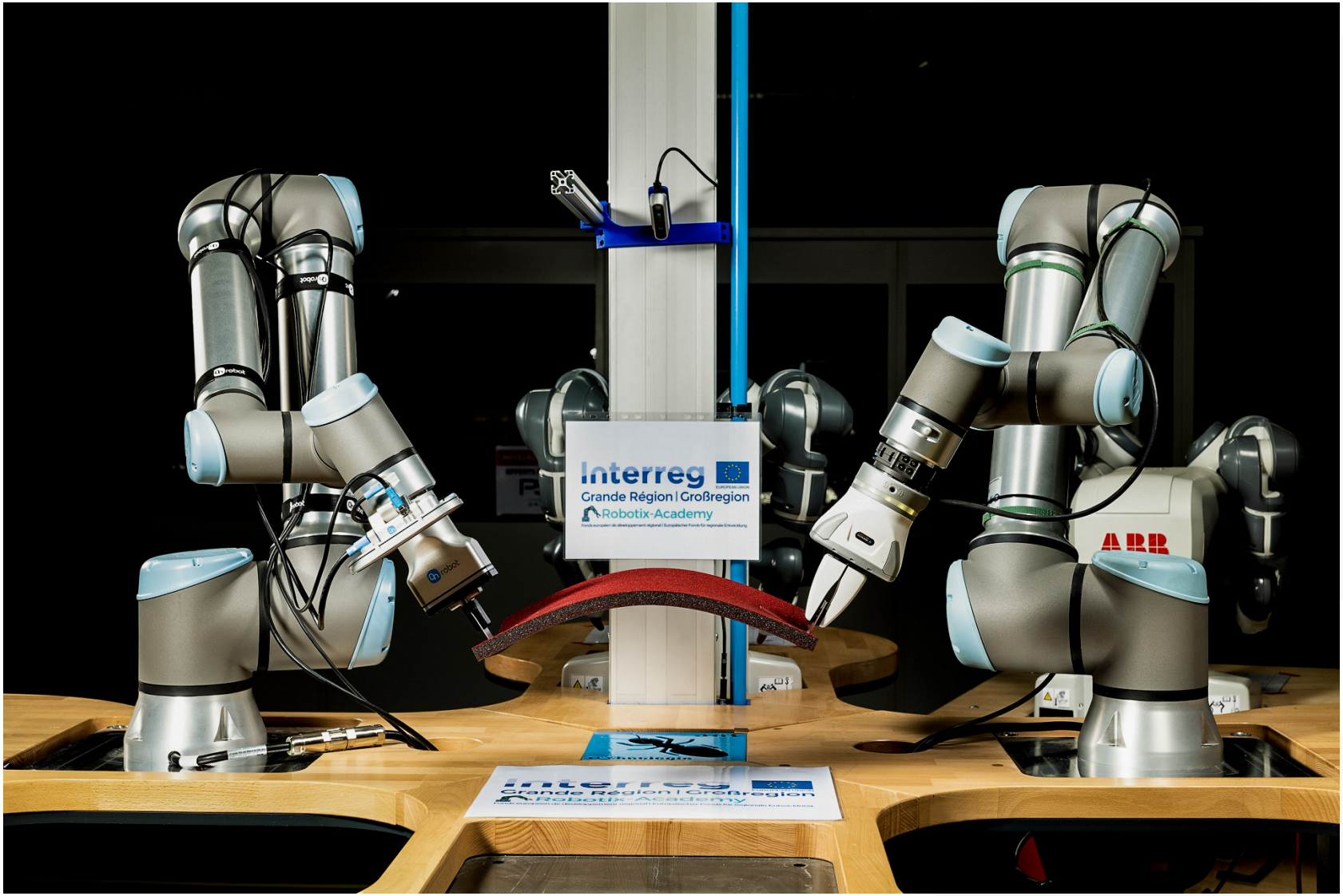
Contribution overview



Main elements:

- Proportional control law
- Visual feedback
- Finite element simulation
- Physical system : robots + deformable object

Experimental setup



Contents

1. Introduction
2. Contribution outline
3. Methods
4. Experiments
5. Conclusion

Control law

Definitions:

- $\mathbf{p}^{(t)}$ - perceived shape
- $\mathbf{e}^{(t)} = \mathbf{p}*^{(t)} - \mathbf{p}^{(t)}$ - shape error
- $\hat{\mathbf{J}}$ - estimated Interaction Jacobian Matrix

Proportional velocity control law such that $\mathbf{e}^{(t)} \rightarrow \mathbf{0}$:

$$\dot{\mathbf{x}}^{(g)} = \hat{\mathbf{J}}_{inv} \mathbf{G}_p \mathbf{e}^{(t)} \text{ with } \mathbf{G}_p > 0$$

Control law

Definitions:

- $\mathbf{p}^{(t)}$ - perceived shape
- $\mathbf{e}^{(t)} = \mathbf{p}*^{(t)} - \mathbf{p}^{(t)}$ - shape error
- $\hat{\mathbf{J}}$ - estimated Interaction Jacobian Matrix

Proportional velocity control law such that $\mathbf{e}^{(t)} \rightarrow \mathbf{0}$:

$$\dot{\mathbf{x}}^{(g)} = \hat{\mathbf{J}}_{inv} \mathbf{G}_p \mathbf{e}^{(t)} \text{ with } \mathbf{G}_p > 0$$

Points of attention:

- Stability analysis wrt. $\mathbf{G}_p$
- Quality of $\mathbf{e}^{(t)}$ estimation
- Quality of $\hat{\mathbf{J}}$ estimation

Control law

Definitions:

- $\mathbf{p}^{(t)}$ - perceived shape
- $\mathbf{e}^{(t)} = \mathbf{p}*^{(t)} - \mathbf{p}^{(t)}$ - shape error
- $\hat{\mathbf{J}}$ - estimated Interaction Jacobian Matrix

Proportional velocity control law such that $\mathbf{e}^{(t)} \rightarrow \mathbf{0}$:

$$\dot{\mathbf{x}}^{(g)} = \hat{\mathbf{J}}_{inv} \mathbf{G}_p \mathbf{e}^{(t)} \text{ with } \mathbf{G}_p > 0$$

Points of attention:

- Stability analysis wrt. $\mathbf{G}_p \rightarrow$ experimental assessment
- Quality of $\mathbf{e}^{(t)}$ estimation
- Quality of $\hat{\mathbf{J}}$ estimation

Control law

Definitions:

- $\mathbf{p}^{(t)}$ - perceived shape
- $\mathbf{e}^{(t)} = \mathbf{p}*^{(t)} - \mathbf{p}^{(t)}$ - shape error
- $\hat{\mathbf{J}}$ - estimated Interaction Jacobian Matrix

Proportional velocity control law such that $\mathbf{e}^{(t)} \rightarrow \mathbf{0}$:

$$\dot{\mathbf{x}}^{(g)} = \hat{\mathbf{J}}_{inv} \mathbf{G}_p \mathbf{e}^{(t)} \text{ with } \mathbf{G}_p > 0$$

Points of attention:

- Stability analysis wrt. $\mathbf{G}_p \rightarrow$ experimental assessment
- Quality of $\mathbf{e}^{(t)}$ estimation $\rightarrow$ replaced by vision feedback
- Quality of $\hat{\mathbf{J}}$ estimation

Control law

Definitions:

- $\mathbf{p}^{(t)}$ - perceived shape
- $\mathbf{e}^{(t)} = \mathbf{p}*^{(t)} - \mathbf{p}^{(t)}$ - shape error
- $\hat{\mathbf{J}}$ - estimated Interaction Jacobian Matrix

Proportional velocity control law such that $\mathbf{e}^{(t)} \rightarrow \mathbf{0}$:

$$\dot{\mathbf{x}}^{(g)} = \hat{\mathbf{J}}_{inv}\mathbf{G}_p\mathbf{e}^{(t)} \text{ with } \mathbf{G}_p > 0$$

Points of attention:

- Stability analysis wrt. $\mathbf{G}_p \rightarrow$ experimental assessment
- Quality of $\mathbf{e}^{(t)}$ estimation $\rightarrow$ replaced by vision feedback
- Quality of $\hat{\mathbf{J}}$ estimation $\rightarrow$ **main concern!**

Jacobian estimation #1

Analytic method. Partition of the nodes of the mesh[10]:

- Grasped nodes g - in contact with the robot
- Target nodes t - defining the target shape to reach
- Free nodes f - none of the above

[10] Koessler et al., ICRA 2021

Jacobian estimation #1

Analytic method. Partition of the nodes of the mesh[10]:

- Grasped nodes g - in contact with the robot
- Target nodes t - defining the target shape to reach
- Free nodes f - none of the above

Linearized quasi-static equilibrium equation:

$$\mathbf{K}(\mathbf{u})\boldsymbol{\Delta}\mathbf{u} = \boldsymbol{\Delta}\mathbf{f} \quad \Rightarrow \quad \begin{aligned} \mathbf{K}_{gg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{gt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{gf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_g \\ \mathbf{K}_{tg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{tt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{tf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_t \\ \mathbf{K}_{fg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{ft}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{ff}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_f. \end{aligned}$$

[10] Koessler et al., ICRA 2021

Jacobian estimation #1

Analytic method. Partition of the nodes of the mesh[10]:

- Grasped nodes g - in contact with the robot
- Target nodes t - defining the target shape to reach
- Free nodes f - none of the above

Linearized quasi-static equilibrium equation:

$$\mathbf{K}(\mathbf{u})\boldsymbol{\Delta}\mathbf{u} = \boldsymbol{\Delta}\mathbf{f} \quad \Rightarrow \quad \begin{aligned} \mathbf{K}_{gg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{gt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{gf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_g \\ \mathbf{K}_{tg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{tt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{tf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_t \\ \mathbf{K}_{fg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{ft}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{ff}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_f. \end{aligned}$$

Assuming constant external efforts, it holds

$$\mathbf{J} = -(\mathbf{K}_{td} - \mathbf{K}_{tf}\mathbf{K}_{ff}^{-1}\mathbf{K}_{fd})^{+}(\mathbf{K}_{tt} - \mathbf{K}_{tf}\mathbf{K}_{ff}^{-1}\mathbf{K}_{ft})$$

[10] Koessler et al., ICRA 2021

Jacobian estimation #1

Analytic method. Partition of the nodes of the mesh[10]:

- Grasped nodes g - in contact with the robot
- Target nodes t - defining the target shape to reach
- Free nodes f - none of the above

Linearized quasi-static equilibrium equation:

$$\mathbf{K}(\mathbf{u})\boldsymbol{\Delta}\mathbf{u} = \boldsymbol{\Delta}\mathbf{f} \quad \Rightarrow \quad \begin{aligned} \mathbf{K}_{gg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{gt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{gf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_g \\ \mathbf{K}_{tg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{tt}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{tf}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_t \\ \mathbf{K}_{fg}\boldsymbol{\Delta}\mathbf{u}_g + \mathbf{K}_{ft}\boldsymbol{\Delta}\mathbf{u}_t + \mathbf{K}_{ff}\boldsymbol{\Delta}\mathbf{u}_f &= \boldsymbol{\Delta}\mathbf{f}_f. \end{aligned}$$

Assuming constant external efforts, it holds

$$\mathbf{J} = -(\mathbf{K}_{td} - \mathbf{K}_{tf}\mathbf{K}_{ff}^{-1}\mathbf{K}_{fd})^{+}(\mathbf{K}_{tt} - \mathbf{K}_{tf}\mathbf{K}_{ff}^{-1}\mathbf{K}_{ft})$$

However, inverting $\mathbf{K}_{ff}$ becomes untractable for finely meshed objects.

[10] Koessler et al., ICRA 2021

Jacobian estimation #2

Simulation method. Computation of the Jacobian using finite differences[11].

For each DOF j of the gripper, a small perturbation $\mathbf{\Delta}[:, j]$ is introduced:

$\mathbf{p}^{(t)} = \text{simulate}(\mathbf{x}^{(g)})$
for $j = 1$ to $\text{length}(\mathbf{x}^{(g)})$ **do**
  $\mathbf{x}_{\Delta}^{(g)} = \mathbf{x}^{(g)} + \mathbf{\Delta}[:, j]$
  $\mathbf{p}_{\Delta}^{(t)} = \text{simulate}(\mathbf{x}_{\Delta}^{(g)})$
  $\hat{\mathbf{J}}[:, j] = (\mathbf{p}_{\Delta}^{(t)} - \mathbf{p}^{(t)})/(\|\Delta[:, j]\|)$
end for

[11] Baksic et al., ICRA 2021

Jacobian estimation #2

Simulation method. Computation of the Jacobian using finite differences[11].

For each DOF j of the gripper, a small perturbation $\mathbf{\Delta}[:, j]$ is introduced:

1: $\mathbf{p}^{(t)} = \text{simulate}(\mathbf{x}^{(g)})$
2: **for** $j = 1$ to $\text{length}(\mathbf{x}^{(g)})$ **do**
3: $\quad \mathbf{x}_{\Delta}^{(g)} = \mathbf{x}^{(g)} + \mathbf{\Delta}[:, j]$
4: $\quad \mathbf{p}_{\Delta}^{(t)} = \text{simulate}(\mathbf{x}_{\Delta}^{(g)})$
5: $\quad \hat{\mathbf{J}}[:, j] = (\mathbf{p}_{\Delta}^{(t)} - \mathbf{p}^{(t)})/(\|\Delta[:, j]\|)$
6: **end for**

The $\text{simulate}()$ method must be run $\text{length}(\mathbf{x}^{(g)}) + 1 = 7$ times to estimate $\hat{\mathbf{J}}$.

[11] Baksic et al., ICRA 2021

Jacobian estimation #2

Simulation method. Computation of the Jacobian using finite differences[11].

For each DOF j of the gripper, a small perturbation $\mathbf{\Delta}[:, j]$ is introduced:

1: $\mathbf{p}^{(t)} = \text{simulate}(\mathbf{x}^{(g)})$
2: **for** $j = 1$ to $\text{length}(\mathbf{x}^{(g)})$ **do**
3: $\quad \mathbf{x}_{\Delta}^{(g)} = \mathbf{x}^{(g)} + \mathbf{\Delta}[:, j]$
4: $\quad \mathbf{p}_{\Delta}^{(t)} = \text{simulate}(\mathbf{x}_{\Delta}^{(g)})$
5: $\quad \hat{\mathbf{J}}[:, j] = (\mathbf{p}_{\Delta}^{(t)} - \mathbf{p}^{(t)})/(\|\Delta[:, j]\|)$
6: **end for**

The $\text{simulate}()$ method must be run $\text{length}(\mathbf{x}^{(g)}) + 1 = 7$ times to estimate $\hat{\mathbf{J}}$.

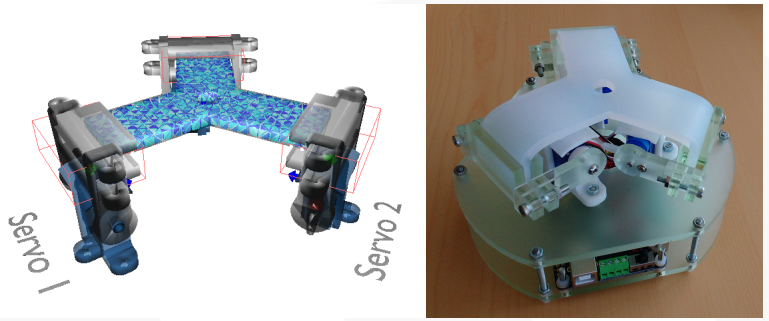
$\rightarrow$ Less "explosive" computation, but high efficiency is still needed!

[11] Baksic et al., ICRA 2021

Time-efficient FE simulation

State of the art:

- Soft robot static analysis[12] and control[13]
- Surgical robotics[14]

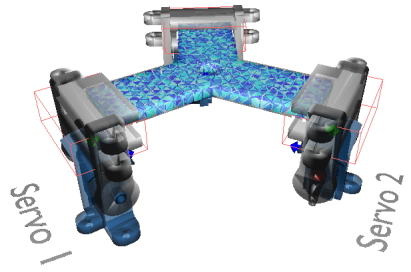


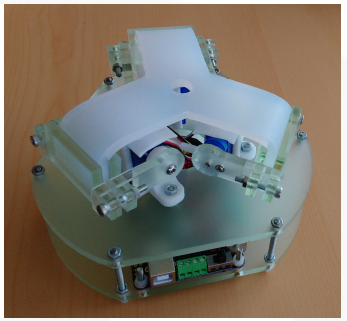
[12] Koehler et al., RA-L 4(2), 2019
[13] Katzschmann et al., RoboSoft 2019
[14] Coevoet et al., RA-L 7(2), 2022
[15] Baksic et al., ICRA 2021

Time-efficient FE simulation

State of the art:

- Soft robot static analysis[12] and control[13]
- Surgical robotics[14]





Software: Bullet, **SOFA framework**

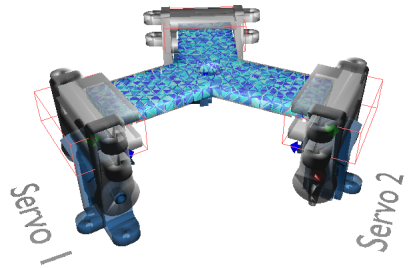
- Parallelized GPU computing enabled
- Support from community (**Inria MIMESIS** and DEFROST teams)

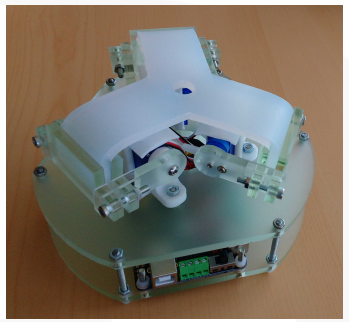
[12] Koehler et al., RA-L 4(2), 2019
[13] Katzschmann et al., RoboSoft 2019
[14] Coevoet et al., RA-L 7(2), 2022
[15] Baksic et al., ICRA 2021

Time-efficient FE simulation

State of the art:

- Soft robot static analysis[12] and control[13]
- Surgical robotics[14]





Software: Bullet, **SOFA framework**

- Parallelized GPU computing enabled
- Support from community (**Inria MIMESIS** and DEFROST teams)

Tricks for fast computation:[15]

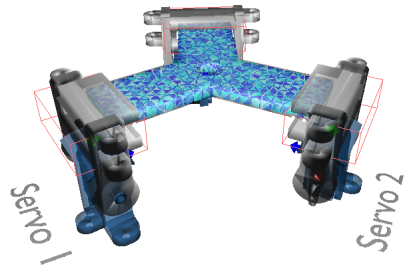
- Parallel computing (LDL decomposition)
- System preconditioning
- Approximated inverse matrices

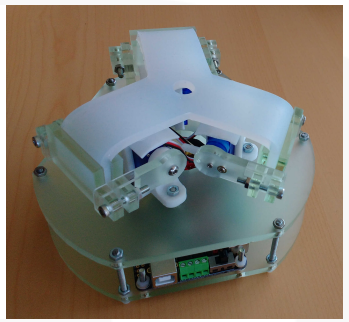
[12] Koehler et al., RA-L 4(2), 2019
[13] Katzschmann et al., RoboSoft 2019
[14] Coevoet et al., RA-L 7(2), 2022
[15] Baksic et al., ICRA 2021

Time-efficient FE simulation

State of the art:

- Soft robot static analysis[12] and control[13]
- Surgical robotics[14]





Software: Bullet, **SOFA framework**

- Parallelized GPU computing enabled
- Support from community (**Inria MIMESIS** and DEFROST teams)

Tricks for fast computation:[15]

- Parallel computing (LDL decomposition)
- System preconditioning
- Approximated inverse matrices

$\rightarrow$ Experimental assessment of $\hat{\mathbf{J}}$ computation frequency

[12] Koehler et al., RA-L 4(2), 2019
[13] Katzschmann et al., RoboSoft 2019
[14] Coevoet et al., RA-L 7(2), 2022
[15] Baksic et al., ICRA 2021

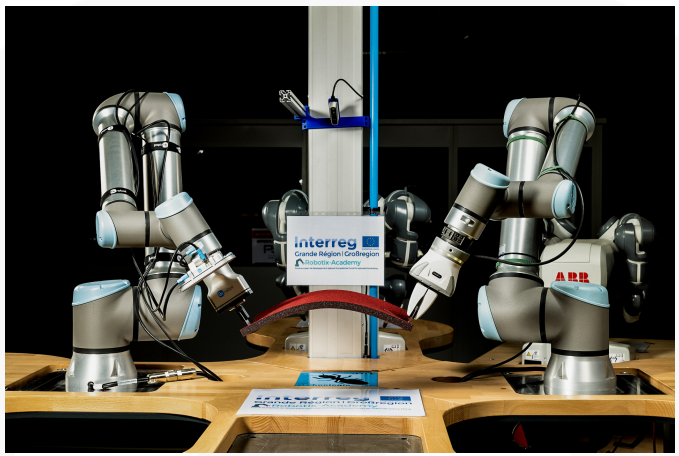
Contents

1. Introduction
2. Contribution outline
3. Methods
4. Experiments
5. Conclusion

Implementation details

Robots:

- 2x UR10e cobots with grippers
- Official ROS driver[16]



[16] https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
[17] https://github.com/sofa-framework/SofaPython3/

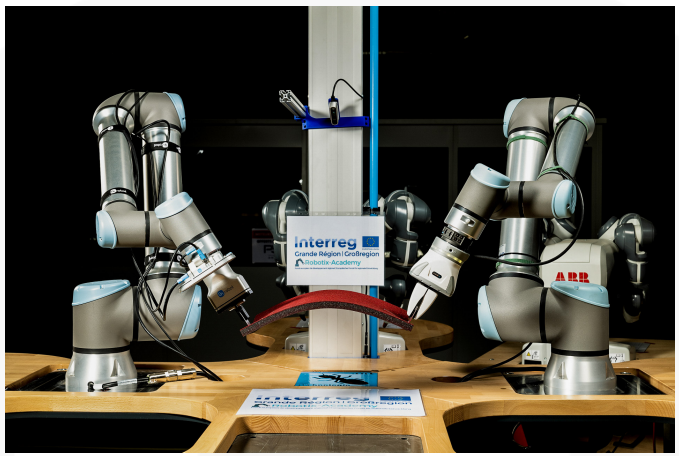
Implementation details

Robots:

- 2x UR10e cobots with grippers
- Official ROS driver[16]

Deformable object :

- Painted foam with adhesive markers
- Tensile testing: $E = 95$ kPa, $\nu = 0.45$.

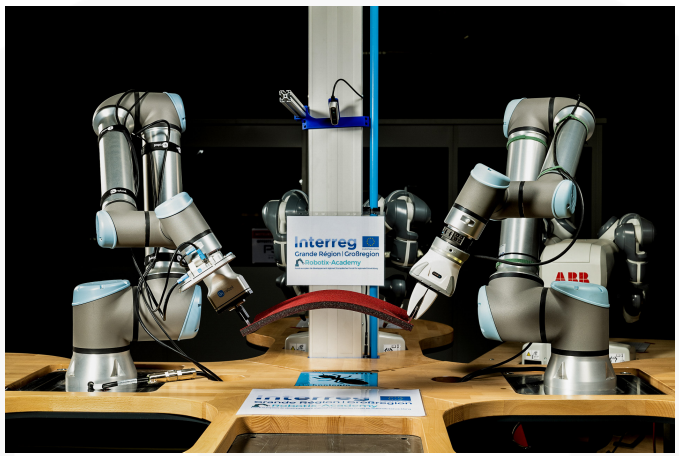


[16] https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
[17] https://github.com/sofa-framework/SofaPython3/

Implementation details

Robots:

- 2x UR10e cobots with grippers
- Official ROS driver[16]



Deformable object :

- Painted foam with adhesive markers
- Tensile testing: $E = 95$ kPa, $\nu = 0.45$.

Vision system:

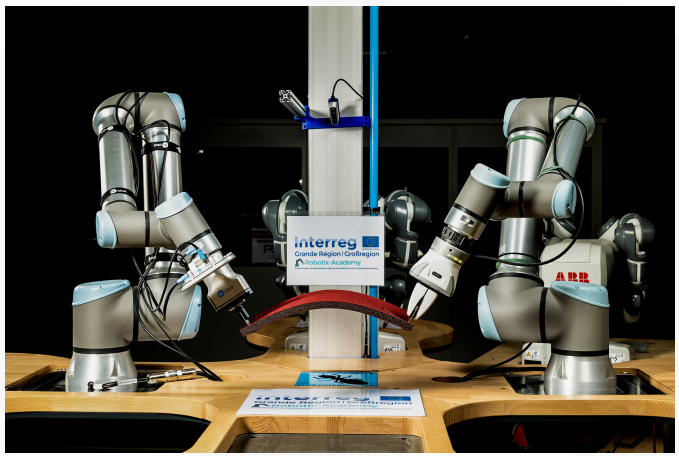
- Intel RealSense D435 camera
- Lucas-Kanade feature tracking (Python + OpenCV)

[16] https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
[17] https://github.com/sofa-framework/SofaPython3/

Implementation details

Robots:

- 2x UR10e cobots with grippers
- Official ROS driver[16]



Deformable object :

- Painted foam with adhesive markers
- Tensile testing: $E = 95$ kPa, $\nu = 0.45$.

Vision system:

- Intel RealSense D435 camera
- Lucas-Kanade feature tracking (Python + OpenCV)

SOFA simulation:

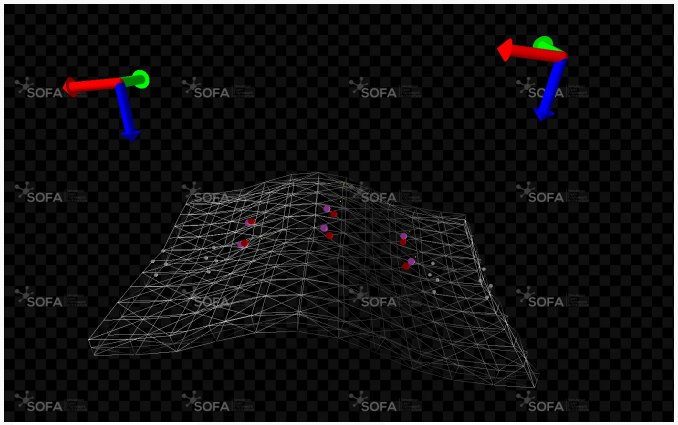
- Co-rotational modeling, linear elasticity, 1452 tetrahedra
- Python scripting[17] + custom libraries

[16] https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
[17] https://github.com/sofa-framework/SofaPython3/

Setup assessment

Modeling errors:

- Coarse mesh, unrealistic deformation at grasping points
- Unmodeled viscosity
- Manual positioning at $t = 0$

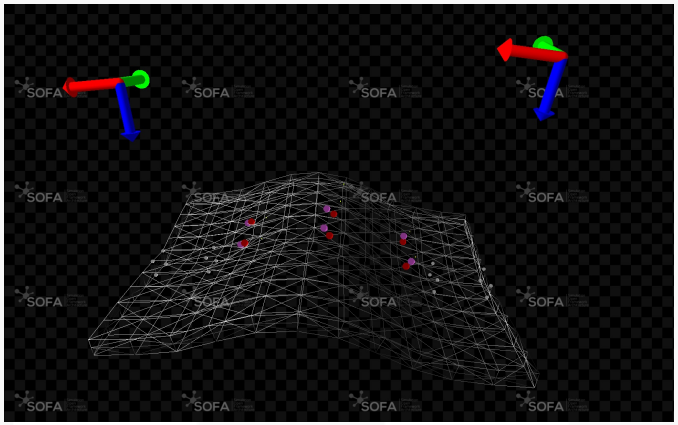


Setup assessment

Modeling errors:

- Coarse mesh, unrealistic deformation at grasping points
- Unmodeled viscosity
- Manual positioning at $t = 0$

$\rightarrow$ looking for vision-based non-rigid registration

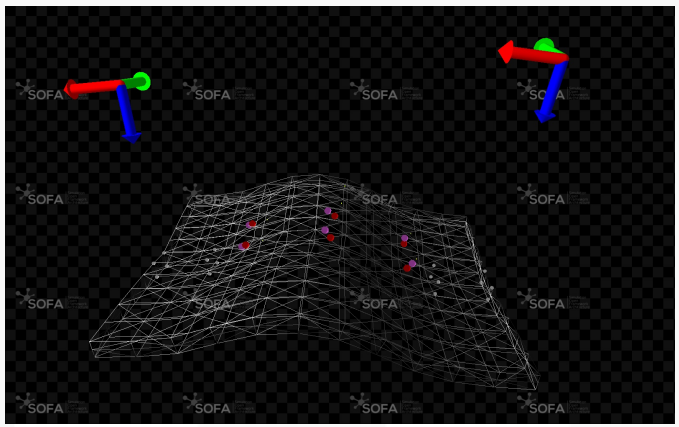


Setup assessment

Modeling errors:

- Coarse mesh, unrealistic deformation at grasping points
- Unmodeled viscosity
- Manual positioning at $t = 0$

$\rightarrow$ looking for vision-based non-rigid registration



Vision system precision:

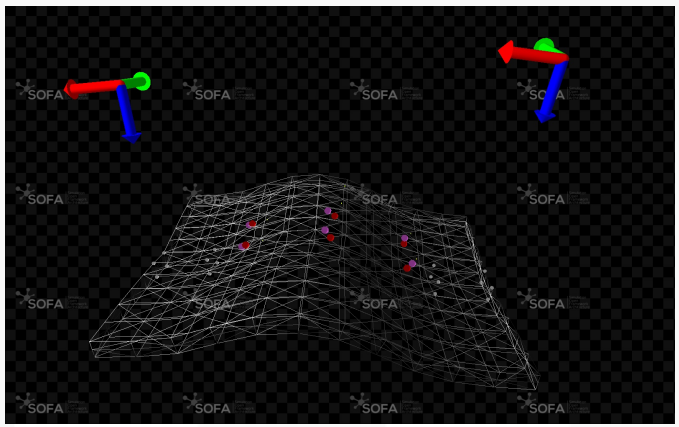
- Depth noise $\pm 1\%$ of distance from camera to object
- Poor repeatability of feature definition (hand-picked)
- After calibration: RMS reprojection error of 6.4 mm

Setup assessment

Modeling errors:

- Coarse mesh, unrealistic deformation at grasping points
- Unmodeled viscosity
- Manual positioning at $t = 0$

→ looking for vision-based non-rigid registration

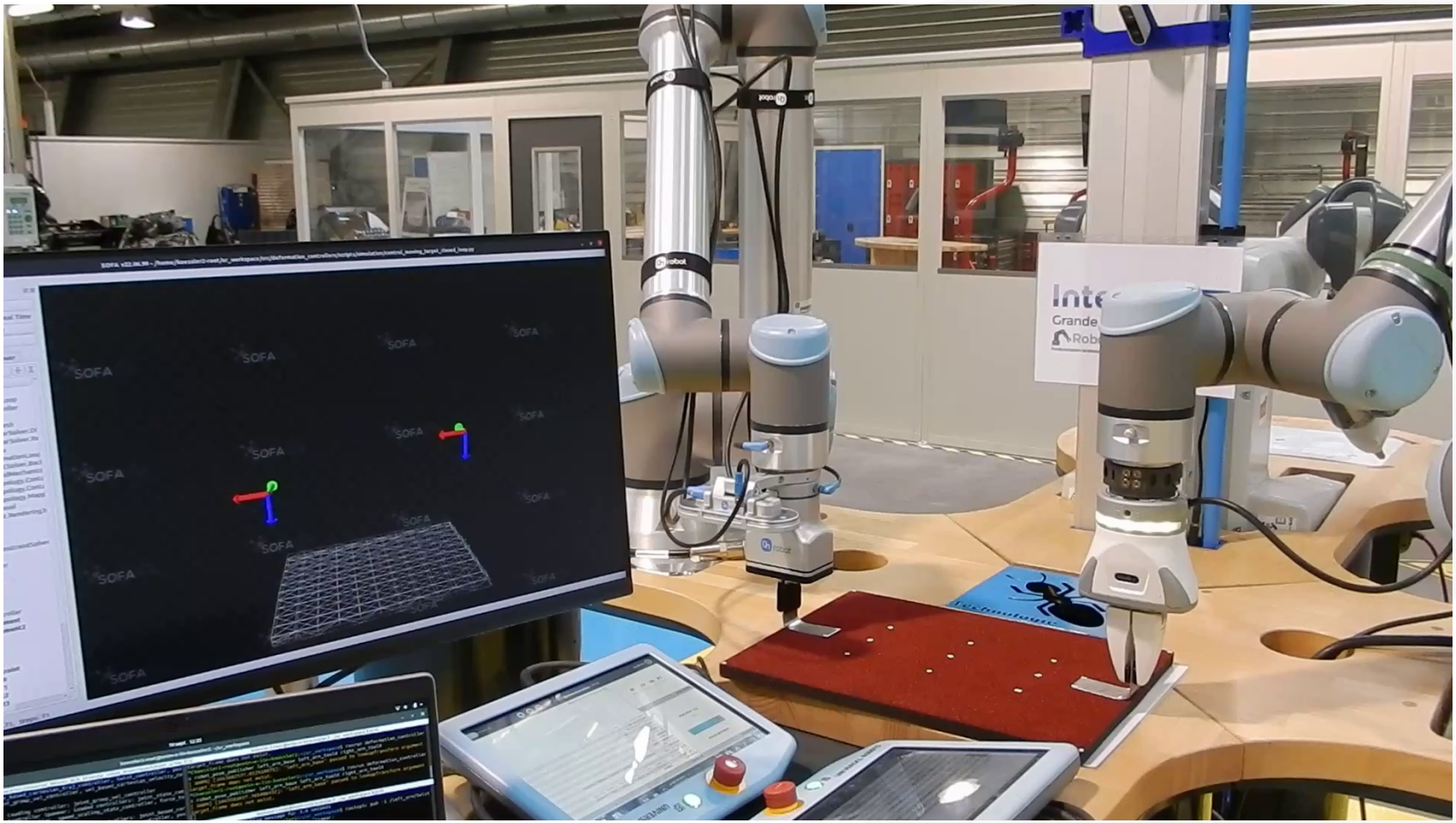


Vision system precision:

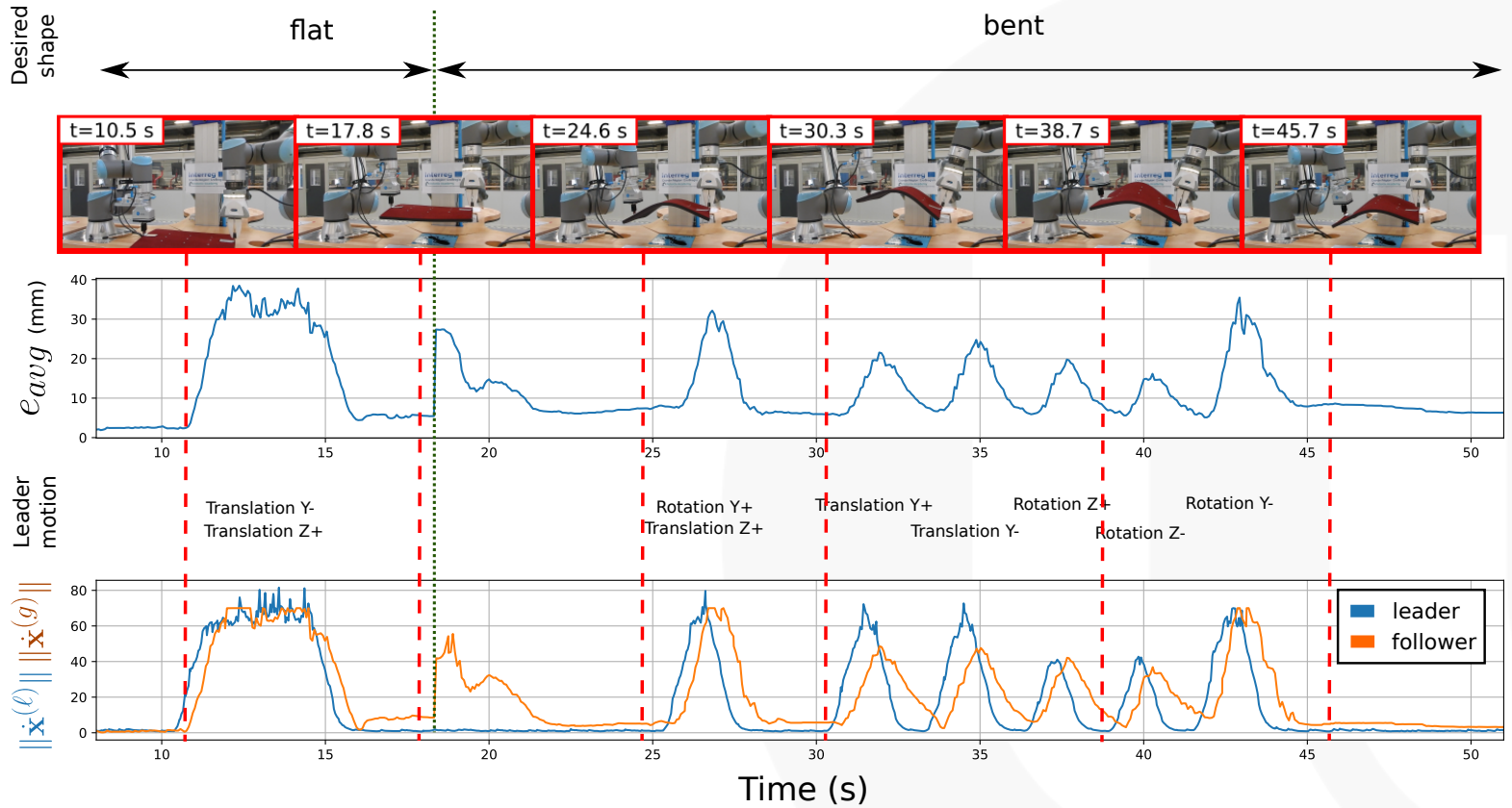
- Depth noise $\pm 1\%$ of distance from camera to object
- Poor repeatability of feature definition (hand-picked)
- After calibration: RMS reprojection error of 6.4 mm

→ Shaping precision cannot be assessed finely
→ Need for a ground truth in the future

Online experiment - Shape recovery along trajectory

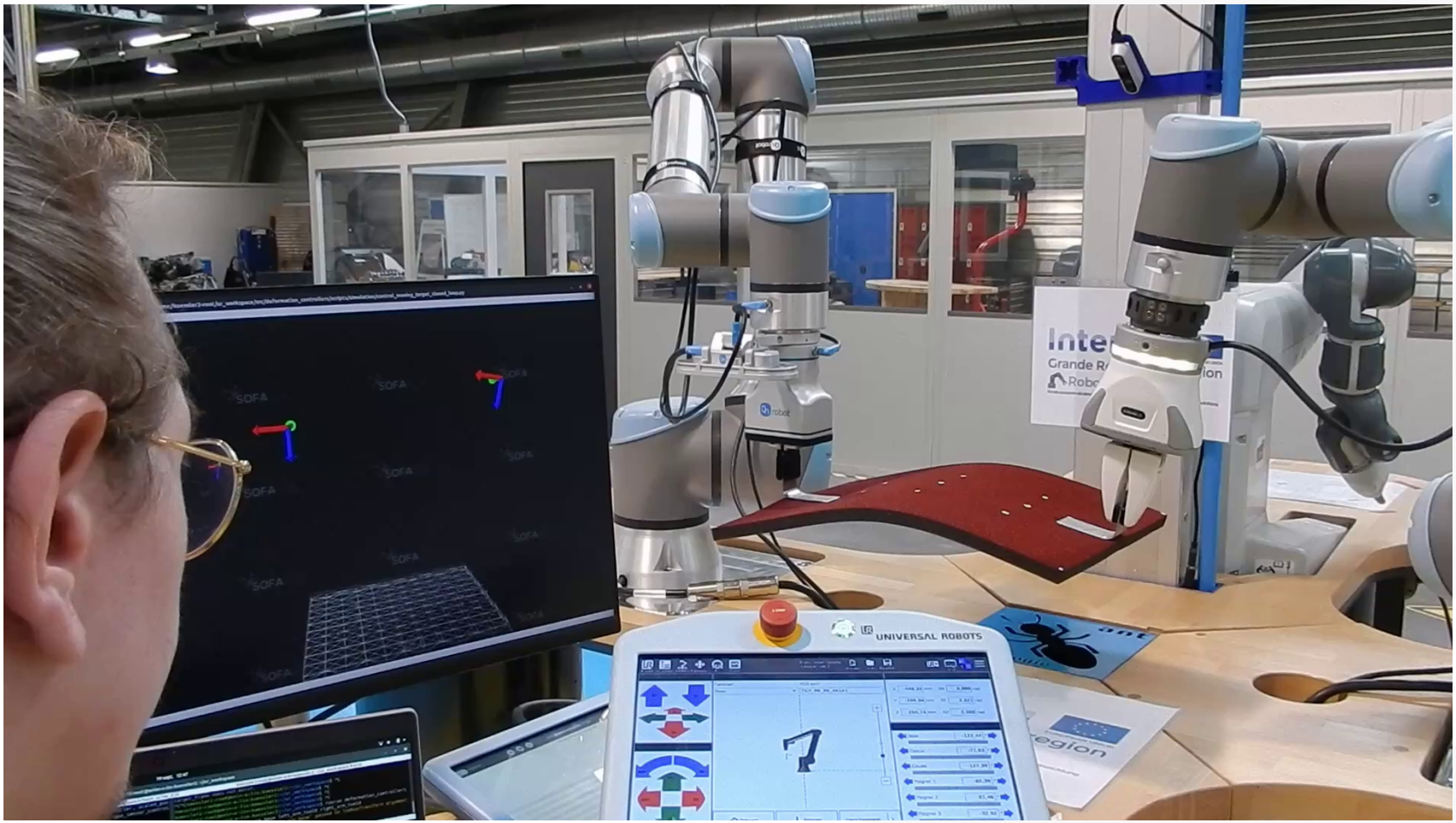


Experimental analysis



- Delay/following error between leader and follower
- Convergence despite model mismatch with shape error < 10 mm
- Steady FPS for simulation (around 15 Hz)

Online experiment - User-controlled master



Contents

1. Introduction
2. Contribution outline
3. Methods
4. Experiments
5. Conclusion

Conclusion and perspectives

Main takeaways:

- Successful hardware and software implementation
- Quality of Jacobian estimation
- Robustness of the control scheme

Conclusion and perspectives

Main takeaways:

- Successful hardware and software implementation
- Quality of Jacobian estimation
- Robustness of the control scheme

Limitations:

- Precision evaluation $\rightarrow$ address vision system limitations
- Need for texture $\rightarrow$ better use of depth and contour information
- Slow dynamics $\rightarrow$ more refined control strategy

Conclusion and perspectives

Main takeaways:

- Successful hardware and software implementation
- Quality of Jacobian estimation
- Robustness of the control scheme

Limitations:

- Precision evaluation $\rightarrow$ address vision system limitations
- Need for texture $\rightarrow$ better use of depth and contour information
- Slow dynamics $\rightarrow$ more refined control strategy

Future directions:

- Multi-robot manipulation ($n \geq 2$)
- Teleoperation in surgical context
- Human-object-robot manipulation

Thank you for your attention!

RoMaDO-SI@

KYOTO 2022 IROS iros2022.org

3rd workshop on RObotic MAnipulation of Deformable Objects: challenges in perception, planning and control for Soft Interaction (ROMADO-SI)

October 23rd, 2022 Kyoto, Japan

https://romado-workshop.github.io/ROMADO2022/

- Cosimo Della Santina, Delft University of Technology, Netherlands, and German Aerospace Center (DLR), Germany
  "to be defined"
- Fanny Ficuciello, Università di Napoli Federico II, Italy
  "Prostate biopsy using a soft/rigid robot"
- Yiannis Karayiannidis, LTH, Lund University, Sweden
  "Manipulation of Deformable Linear Objects: From simulation to real world experiments"
- Jeannette Bohg, Stanford University, USA
  "Real2sim for Deformable Objects"
- Xiang Li, Tsinghua University, China
  "Global Model Learning for Large Deformation Control of Elastic Deformable Linear Objects: An Efficient and Adaptive Approach"
- Doganay Sirintuna, Istituto Italiano di Tecnologia, Italy
  "Human-Robot Collaborative Carrying of Deformable Objects"
- Daniele Meli, University of Verona, Italy
  "Soft tissue manipulation in robotic surgery"
- Miguel Aranda, Universidad de Zaragoza, Spain
  "Shape control tasks in deformable object manipulation"
- Adrien Escande, Japanese-French Joint Robotic Laboratory (JRL), Tsukuba, Japan
  "to be defined"